

CHECKERS in the last issue had one typo; in line 1220 where part of the line read:  
 $\text{@}((\text{S}+\text{R})+2)=3;$  and it should have read:  $\text{@}((\text{S}+\text{R})\text{;}2)=3;$   
 An error in line 8 had too many zeros at 30000.  
 I've had many comments on this program, all pleased with the effect and operation  
 (once the glitch was cleared up).

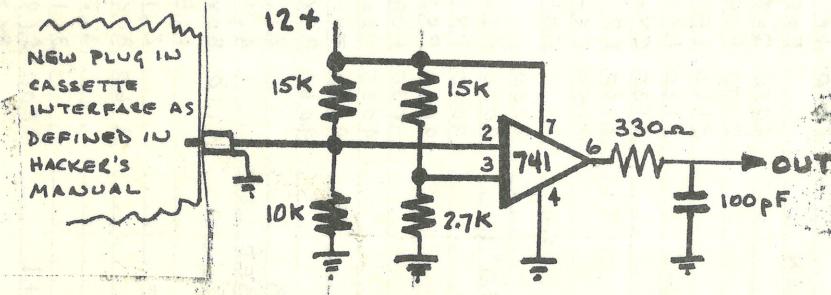
PROOFREADING of listings is getting better, and hopefully I'm not going to have  
 any more problems as I am now sending a copy of the ready-to-print listing to the  
 author for his last minute check.

GIANT LETTERS program on p.45 can be 'clarified' by replacing line 105 with the  
 statement  $\text{X}=\text{0};$  GOSUB C . The zero will stop the machine's printing after it  
 finishes the AN of ARCADIAN. What is happening is - the machine has been set into  
 a printing mode and it keeps on going until it hits an internal halt. The  $\text{X}=\text{0}$  sets  
 such a halt thru the POKE function, where you want it.

CONNECTOR for the 50-pin expansion port is a standard printed circuit device, with  
 25 double-sided pins on 0.1 inch centers. I understand that the APPLE II uses this,  
 as well as the SHUGART disc drive. There are some of these on the surplus market  
 here, 3M No. 3415, and I can supply these at \$2.50 ppd, including a 3" bit of flat  
 cable that you could solder to.

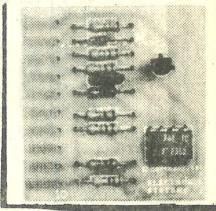
KEYBOARD project in work utilizes a CHERRY brand (?) keyboard wired in parallel  
 with the keypad with some buffers so that either can be used. All letters and  
 characters are where they belong, while the shift key allows the generation of the  
 various WORDS. The Keyboard is the one advertised by JAMES ELECTRONICS, 1021  
 Howard Ave., San Carlos CA 94070, at \$29.95. "63 Key Uncoded Keyborad".

PRINTER project has been made to work ,using a Type 43 Teletype machine and a kit  
 for an interface available from ELECTRONIC SYSTEMS, P.O. Box 21638, San Jose, CA  
 95151, their part #232A at \$7. plus postage. I saw an ad of theirs in KILOBAUD,  
 April, p.172. The schematic of this kit is included for you scratch-builders:



**RS-232/ TTL\*  
INTERFACE**

- Converts TTL to RS-232, and converts RS-232 to TTL
- Two separate circuits
- Requires -12 and +12 volts
- All connections go to a 10 pin gold plated edge connector
- Board only \$4.50
- Part No. 232, with parts \$7.00 Part No. 232A



COLOR STANDARDS question came up, and I find that Bally would make an excellent  
 color generator, as apparently their color output is very well controlled. I  
 wonder if one of the TV-technical types can compare colors with a standard color  
 generator output and tell us which color numbers give you the standard shades  
 necessary for color TV work.

BANGMAN game program included this month is by Ernie Sams, 248 S. Forest Street,  
 Bellingham WA 98225. It has a novel twist over the old hangman game and is quite  
 clever. It has a good scheme for entering characters without their appearing on  
 the screen, and a search routine that can locate and account for multi-usage of  
 a letter. I am also including Ernie's sheet of documentation that will be of help  
 to a lot of us.

Line #	Statement(s)	Line #	Statement(s)	Line #	Statement(s)
3	BANGMAN	14.3.0	$G = (KN(1.) + 13.2) \div 14.0 + 6.5$	9.22.5	FOR U=Φ TO 5
5	BY E. SAMS (C) 3-3-79	1.44.0	$CY=2.3; CX=-7.4; JV=G$	9.23.0	$\&(2.3)=2.55; \&(2.1)=2.55$
7		1.45.0	IF TR(1.)=Φ GOTO 14.3Φ	9.24.0	$\&(2.3)=Φ; \&(2.1)=3.1$
9	NT=2	1.54.0	1.F@((G-4.5)=-1. PRINT "HAS	9.26.0	FOR V=5 TO 2Φ; &(1.9.)=Y; NEXT V
10	CLEAR; PRINT "ONE PERSON	BEEN USED"; GOTO 14.0Φ	9.27.0	$\&(2.1)=Φ; \&(1.9.)=Φ$	
10	KEYS IN A WORD, TO 1Φ, LET- TERS; AND OTHER, TRIES, TO	1.55.0	$@((G-4.5)=-1; CY=-4.0; CX=$	9.30.0	X=RND(1Φ)+5.5
11	GUESS, IT, WITH NO	7.8+((G-6.5)*6); JV=G	9.31.0	Y=RND(1.2)	
12	PRINT "MORE THAN 9, WRONG	1.6.0	FOR C=Φ TO A-1	9.32.0	BOX X <sub>3</sub> Y <sub>3</sub> 1.9.1, 2
13	GUESSES, USING, KN & TR(1.)	1.7.0	CY=-2.5	9.34.0	NEXT U
14	PRINT; PRESS, GO	1.8.0	1.F. G=@(C). CX=-7.9+((C*8)+3); JV=G	9.35.0	$F.C=8Φ; \&(1.9.)=Φ$
15	I=KP; IF I=13, CLEAR	1.9.0	$\&(G; Q=1; @((C)=-1$	9.36.0	BOX 6Φ, 1Φ, 3Φ, 6Φ, 2
16	E=Φ; Q=Φ	2.0.0	NEXT C	9.37.0	BOX -8, 4, 3Φ, 1Φ, 2
17	PRINT "ENTER WORDS, THEN	2.0.0	IF Q≠1 GOSUB 9.6ΦΦ+((E*(1Φ);	9.40.0	FOR Y=Φ TO 17.6; &(1.Φ)=Y; NEXT Y
18	PRESS, GO, TO 9	2.0.50	$E=E+1; IF E=9, GOTO 9.6ΦΦΦ$	9.41.0	$\&(9)=5Φ; F.C=Φ$
19		Q=Φ	9.50.0	$C.X=-5Φ; CY=1.6; PRINT "PRESS,$	
20	FOR A=Φ TO 9	2.1.0	FOR F=Φ TO A-1	9.60.0	GO"; GOTO 5Φ
21	C.Y=3.1	2.1.10	1.F. @((F) # -1. GOTO 2.2ΦΦ	9.60.0	BOX 6Φ, 2.7, 1.9, 1.5 BOX 5Φ,
22	PRINT "LETTER. #", #3, A+1	2.1.20	NEXT F	2.8, 2.9, 1.9, 2Φ BOX 6.2, 2.8, 2.9, 1.9, 2Φ	
23	BOX -5.5, 2.3, 5Φ, 8, 2.	2.1.30	CY=1.7; CX=-5Φ	BOX 6Φ, 2.4, 9, 1.9, 2Φ RETURN	
24	K=KP	2.1.40	PRINT "CONGRATULATIONS	9.61.0	BOX 6Φ, 1.8, 4, 4.5, 1Φ RETURN
25	1.F. K=1.3, GOTO 7.9Φ	2.1.50	CY=Φ; CX=-ΦΦ; PRINT "PRESS,	9.62.0	BOX 6Φ, 6Φ, 1.6, 2Φ, 1Φ RETURN
26	4.20, 1.F. K>9Φ, PRINT "INVALID"	2.2.00	GO"; GOTO 5Φ	9.63.0	BOX 5Φ, 1.4, 4, 3, 1Φ BOX 4.9, 6,
27	Q.0. 31Φ	2.2.00	NEXT D	3.9Φ, 1.9, 1.5, 1Φ RETURN	
28	1.F. K<65, PRINT "INVALID";	9.0.00	CY=-8Φ; CY=-9	9.64.0	BOX 7Φ, 1.4, 4, 3, 1Φ BOX 7Φ, 6Φ
29	Q.0. 31Φ	9.0.10	PRINT "SORRY, ABOUT THIS	9.65.0	3.9Φ, 1.9, 1.5, 1Φ RETURN
30	Q.(A)=K; @((A+1Φ)=K,	9.0.20	PRINT "THE WORD WAS	9.66.0	BOX 5.7, -1.2, 3, 1.5, 1Φ RETURN
31	7.00, NEXT A	9.0.30	FOR H=F TO A-1	9.66.0	BOX 6.4, -1.12, 3, 1.5, 1Φ RETURN
32	7.9Φ, CLEAR	9.0.35	CY=-7.9+((H*8)+3)	9.67.0	BOX 5.5, -2.0, 8, 2Φ, 1Φ RETURN
33	FOR B=F, Φ, TO A-1	9.0.40	IV=@((H+1Φ); NEXT H	9.68.0	BOX 6.7, -2.0, 8, 2Φ, 1Φ RETURN
34	Y=-3.2; X=-8Φ+(B*8)+3	9.0.6.0	BOX -1.6, 4, 1Φ, 1; BOX -8, 7, 9	NOTE: ELIMINATE AS MANY SPACES	
35	10.1.0 BOX X, Y, 7, 3, 1; NEXT B	2.7, 5, 9, 1; BOX -1.9, 1.1, 1; 2, 1; BOX	A.5 POSSIBLE (EXCEPT THOSE		
36	1.1.00 CY=-2.4; CX=6; PRINT "USED LET	4, 1.1, 1, 1, 2, 1; 5, 1, 1; BOX -9, 4, 1, 1; 4, 1,	INSIDE QUOTES). IF YOU RUN		
37	TERS"; CY=-3.2; CX=4.2;	1; BOX -1.8, 7, 3, 3, 2; 2, 1; BOX -8, 6, 1,	SHORT, CUT LINES, 3+30		
38	PRINT "BELOW:"	2, 1, 1; 1; BOX -1.4, 7, 3, 3, 2,			
39	12.00 FOR M=Φ TO 25; @((M+2Φ)=	1.1.6.0 BOX -1.4, 7, 1, 1, 1; BOX -1.1, 1, 1; 7, 9,			
40	M+6.5; NEXT M	3, 2, 2; BOX -9, 6, 2, 1; 2, 1; BOX -7, 8, 8,			
41	13.00 FOR D=Φ TO 25	3.9, 1, 2; BOX -7, 7, 1, 1, 2, 1; BOX -1.7, 9, 2			
42	CY=3.1; CX=-8Φ	6. (9)=2.22Φ; &(d)=8.7; &(1)=8.7Φ			
43	14.10 PRINT "GUESS #", #3, D+1	6. (2)=8Φ; &(3)=8Φ			
44	14.20 BOX -3.3, 1.9, 9, 4, 1.6, 2				

40

**BANGMAN DOCUMENTATION:**

10 - 200 Initialize and instructions.  
 300 - 700 Ask for and accept up to ten valid letters.  
 600 Stores the word in two locations:  
     1. To keep track of the letters correctly guessed.  
     2. To print the word if not guessed in nine tries.  
 800 - 1010 Set up blanks for the word.  
 1200 Store each letter of the alphabet for future use so the same letter is not used twice.  
 1300 Initiate guessing loop. Allows 26 guesses.  
 1430 - 1450 Allows one to guess a letter by turning knob #1.  
 1540 If storage position is set to -1 the letter has been used.  
 1550 Sets value of storage position to -1 and prints letter at the bottom of the screen.  
 1600 - 1900 Loop through the storage positions in 600 (1). If a match is found print the letter in the appropriate location(s) on the blanks established in 800-1010 and change the storage position value to -1. Set flag "Q" to 1.  
 1800  
 2000 If the flag "Q" is not equal to 1 then the chosen letter did not match a letter in the word so go to subroutine 9600 plus counter E times 10 and print that portion of the man. Increment the E counter. If there have been nine wrong guesses default to 9000 to "bang" part of bangman. Otherwise flag "Q" equals 0.  
 2100 - 2200 Loop through storage positions in 600 (1). If all positions are -1 then the word has been guessed. Print "congratulations" and press go to start a new game.  
 9000 - 9020 The word was not guessed within the nine wrong guesses allowed. The man was completely built, so - -  
 9030 - 9040 Print out the word (from 600 (2)) on the blanks.  
 9060 - 9160 Draw a gun with the word COLT on it.  
 9200 Change the screen to a border format.  
 9225 - 9340 Put six random shots in the body of the man. Use sound effects. 9230-9240 is the shot. 9260 is the ricochet.  
 9350 Blank out screen  
 9360 - 9370 Blank out man  
 9400 Uncover screen from top to bottom with man gone.  
 9410 Restore screen to full screen format.  
 9500 Press go to start a new game.  
**SUBROUTINES**  
 9600 Prints head, eyes, mouth.  
 9610 Prints neck.  
 9620 Prints body.  
 9630 Prints right arm.  
 9640 Prints left arm.  
 9650 Prints right leg.  
 9660 Prints left leg.  
 9670 Prints right foot.  
 9680 Prints left foot.

RANDOM ART has been expanded with some added sound and shape enhancements by Dave Stocker. Add the following to last month's program:

15 &( $\emptyset$ ) =7; &(1)=7; &(9)=84; NT= $\emptyset$ , &(21)=14; &(22)=255  
85 B=FC+RND(31)x8+4; &(2)=B; &(3)=B  
125 &(19)=X; &(18)=Y  
135 IF &(23)=1 RUN  
145 Press GO to set new parameters. (try 5,13)

Press GO to set new parameters. (try 5,13)

TUTORIALS by Steve Walters (556 Langfield, Northville, ME 48167) and Dave Iback (19553 Dartmouth) follow. The first provides some very interesting basic data on character size and how to make them appear exactly where you want them. The second provides additional comments on the IF-type statements discussed on p. 41.

## CHARACTER SIZE AND PRINT LOCATION

Every character is 5 pixels wide, with a one-pixel space added to the right of the character to provide a one-pixel separation between characters. Thus, the effective width of a character is 6 pixels.

Every character is 7 pixels high, with a one-pixel space added beneath it to provide a one-pixel separation between lines of print. Thus, the effective height of a character is 8 pixels.

The cursor is displayed as a 6-pixel wide by 8-pixel high box, the effective size of a character. When a program is running, the cursor is not visible. When a program ends, the cursor is displayed wherever it happens to be, preceded by the line entry indicator >.

If a PRINT command is not ended with a comma, the computer will leave a full space (6 pixels by 8 pixels) following the last character in the PRINT statement, and shift to the next print line. This end-of-statement space will appear as a white box against a black background, and will blank out anything located beneath it.

If a PRINT command is ended with a comma, no space is added beyond the one-pixel space to the right of the last character, and the cursor remains at that location until another PRINT command is given.

A character prints centered on its CY location, but not centered on its CX location. Given CX and CY as the print location of a character, the horizontal center of the character is CX-1 and the vertical center of the character is CY. The left edge of the character is located on CX-3 and the right edge of the character is located on CX+1. The top and bottom of the character are located on CY+3 and CY-3, respectively.

Since the left edge of the screen display area is on CX=-80, and the left edge of a character is on CX-3, a character will print at CX=-77 even if the program specifies CX=-78, -79 or -80 prior to the PRINT command. However, this behavior is not duplicated at the right side of the screen. If CX is specified at +78, the right edge of the character will be on CX=+79 (CX+1, and the right limit of the screen display area); but if CX is specified at +79, the character will print beyond the CX=+79 limit. The cursor will shift lines in the process, and if a comma follows the PRINT command, the cursor will shift to CX=-77 on the same CY line.

## SCREEN DISPLAY AREA RELATED TO CHARACTER SIZE

The edge limits of the screen display area (CX=-80 to +79; CY=43 to -44) are functionally related to the character print size. The normal top line of print (without a CY value being specified) is CY=40, and thus the top of the characters on that line are at CY=43. There are 11 normal print lines, located at CY=40, 32, 24, 16, 8, 0, -8, -16, -24, -32 and -40. The bottom line (CY=-40) results in the bottom of the characters on that line being at CY=-43 and the one-pixel space below them being at CY=-44.

Similarly, the screen display area width is 160 pixels. This would allow 26 characters (20x6=156 pixels) with 4 pixels to spare. However, the first line of an entry starts with the line entry indicator > as the first character and so only 25 characters and spaces can be entered. When 25 additional characters have been entered, the one-pixel space to the right of the last character is at CX=75; an additional character cannot fit in the remaining 4 pixels so the cursor shifts to the next line. On second and subsequent lines of an entry, 26 characters can be entered per line.

#### BOX COMMANDS RELATED TO PRINT LOCATION

The dimensions of a box which will outline a printed statement, or blank it out, can be readily determined from the CX and CY values of the PRINT command.

Tracing a printed statement or blanking it out with a reverse box. The smallest box which can frame a character is a reverse box:

```
BOX CX-1, CY, 7, 3, 3
```

The above box is a rectangle which is 7 pixels wide and 9 pixels high superimposed over the 5 pixel by 7 pixel character.

A more esthetically pleasing reverse box frame is obtained from a square box:

```
BOX CX-1, CY5, 7, 2
```

A similar formula can be used to blank out a character:

```
BOX CX-1, CY, 9, 9, 3
```

Note that the reverse box must be produced after the PRINT command for the statement being framed or blanked out has been executed.

Tracing a printed statement with an outline box. The outline box X is more useful than the reverse frame box because the outline box can be produced before the PRINT command is executed, and thus the character can be printed and reprinted inside the outline box as desired during a program.

The smallest outline box which can frame a character is:

```
BOX CX-1, CY, 9, 11, 1
```

```
BOX CX-1, CY, 7, 9, 2
```

Again, a square outline box is more esthetically pleasing:

```
BOX CX-1, CY, 11, 11, 1
```

```
BOX CX-1, CY, 9, 9, 2
```

Outline box for a statement of more than one character. Given CX and CY as the PRINT start location for a row of  $n$  characters ( $n=1$  or more), with a comma "n" following the PRINT instruction, the outline box which will frame the row of "n" characters can be determined from the following:

```
BOX CX-1+3(n-1), CY, 11+6(n-1), 11, 1
```

```
BOX CX-1+3(n-1), CY, 9+6(n-1), 9, 2
```

Note: a reverse box could also frame the row of characters using either of the above BOX commands with a 3 as the ending code.

If a comma is not desired because of the 6-pixel space added after the last character, a larger outline box is required:

```
BOX CX-1+3(n-1), CY, 21+6(n-1), 11, 1
```

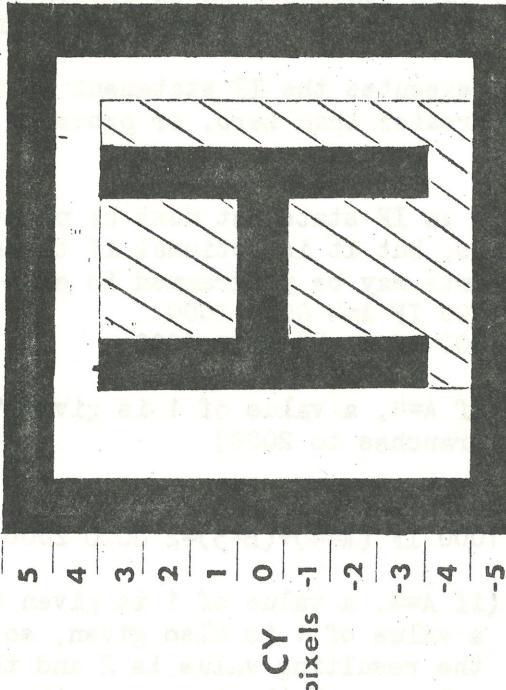
```
BOX CX-1+3(n-1), CY, 19+6(n-1), 9, 2
```

Illustration of an outline box and a printed character inside it:

```
BOX CX-1, CY, 11, 11, 1
```

```
BOX CX-1, CY, 9, 9, 2
```

```
CX=0; CY=0; F=INT, "H"
```



-6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |

CX pixels

Outline box for a statement of more than one character. Given CX and CY as the PRINT start location for a row of  $n$  characters ( $n=1$  or more), with a comma "n" following the PRINT instruction, the outline box which will frame the row of "n" characters can be determined from the following:

-6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |

CX pixels

# arcadian

TUTORIAL 2 is a follow-up on last issue's comments on IF statements, again by Dave and Steve.

The Bally BASIC interprets IF statements in terms of Boolean algebra concepts. In simple terms, each condition in an IF statement is assigned a value of one (+1) if it is true (i.e., if it is met) or a value of zero (0) if it is false (not met).

The program then executes the IF statement if the resulting Boolean value of the IF statement is greater than zero, or proceeds to the next program line if the value is zero.

Each condition in an IF statement must be placed in parentheses if there are two or more conditions, but it is optional if there is only one condition. Thus, simple IF statements may be programmed in either of two ways:

1000 IF A=4 GOTO 2000	(uses 12 bytes)
1000 IF (A=4) GOTO 2000	(uses 14 bytes)

(if A=4, a value of 1 is given to the IF statement and the program branches to 2000)

1000 IF A=4 IF B=5 GOTO 2000	(uses 16 bytes)
1000 IF (A=4)+(B=5)=2 GOTO 2000	(uses 22 bytes)

(if A=4, a value of 1 is given to the IF statement, and if B=5, a value of 1 is also given, so that if both conditions are true the resulting value is 2 and the program branches to 2000; if only one of the two, or neither, is true, the program does not branch)

Obviously, in both of the above cases, using the parentheses costs more in memory bytes and is thus not efficient. However, in more complicated IF statement conditions, parentheses result in less memory bytes being used; and in some cases the saving is substantial.

For example, consider a condition where A=4 or B=5 or both (i.e., A=4 and/or B=5):

1000 IF A=4 GOTO 2000	
1010 IF B=5 GOTO 2000	(uses 24 bytes)
1000 IF (A=4)+(B=5) GOTO 2000	(uses 20 bytes)

Even more memory savings would be involved if the condition was any one or more of three variable values.

The following list illustrates other applications of special IF statements in the most byte-efficient form, based on the above principals:

## Special IF statements:

If A=1 and/or B=2	1000 IF (A=1) + (B=2) GOTO 2000
If A=1 and B=2	1000 IF A=1 IF B=2 GOTO 2000
If A=1 or B=2 but not both	1000 IF (A=1) - (B=2) GOTO 2000
If A ≠ 0 (i.e., if ABS (A) ≠ 0)	1000 IF A GOTO 2000
This is also useful for hand control responses:	
1000 IF TR(1) GOTO 2000	will branch if trigger is pulled
1000 IF JX(1) GOTO 2000	will branch if joystick moved sideways
1000 IF JY(1) GOTO 2000	will branch if joystick moved back & forth
If neither A nor B equal zero	1000 IF A IF B GOTO 2000
If A=1 and B=2 and C=3	1000 IF A=1 IF B=2 IF C=3 GOTO 2000
If A=1 and/or B=2 and/or C=3	1000 IF (A=1) + (B=2) + (C=3) GOTO 2000
If A=1 or B=2 or C=3 but not more than one of these	1000 IF(A=1) + (B=2) + (C=3) =1 GOTO 2000
If any two of the above three conditions are to be met	1000 IF(A=1) + (B=2) + (C=3) =2 GOTO 2000

TUTORIAL 3 is another follow-up, this time by Jean Taillefer.

The >= and =< symbols will work in an IF statement. They stand for "equal to or greater than" and "less than or equal to", respectively. The symbols are sometimes shown as ≥ or ≤.

The statement IF A>=B GOTO 100 is equivalent to

IF A>B GOTO 100; IF A=B GOTO 100      or  
IF(A>B) + (A=B) GOTO 100

A statement that is equivalent to the 'reverse sign' function on a calculator can be accomplished with the following: -(A)

If A is positive, the statement is evaluated as -(+A), for a negative number.

If A is negative, the statement is evaluated as -(-A), for a positive number.

This statement can be used with a variety of other statements, such as:

PRINT -(A)  
B=-(A)  
IF -(A) = 1 GOTO 100

This is different from the ABS command which always gives a positive number.

DEALER in the St. Louis area is RTS Electronics/Gametronics, 356 Brookes Dr. 731-2309 who report that they have all cartridges in stock, do mail order, and hold Football tournaments for players on Tuesdays, 6-9pm and Saturdays, noon to 4. They mention that players are bringing their own hand controllers, which makes me think of the slot car operators...

DEALER in Indiana is ABC Hobbycraft 2155 E. Morgan Ave, Evansville 47711 (477-9661) who are open Sundays, and who act as the local hangout for Bally enthusiasts.

GAMES 2007 PINBALL and 2006 SPACE INVADERS are scheduled for July release. There were a few words on these on p. 23.

STATEMENT When I started this paper late last year, I indicated in my subscription form that I would print material as it became available, but at least bi-monthly in this fiscal year. Well, I have been doing better than bi-monthly because material has been coming in at a much higher rate than I expected, so I have been publishing more or less monthly. This is a hobby output so publication is bound to be somewhat haphazard. Post-printing operations are starting to get unwieldy and I may have to get professional help in the folding/stapling area. Right now it looks like there will be at least 2 more issues to November, so you'll be getting at least a 50% 'profit'.

ADS Sell: Bally ARCADE with 4 hand controls, 280ZZZAP, Baseball, Tiny BASIC. First certified check for \$260. R.Dermody, 8431 Timber Glen, San Antonio, TX 78250

A fantasy game package is available for those who enjoy the Dragon/Dungeon type of operation. G.McLimore, 1210 E.Virginia St. Evansville IN 47711. Games are MULTIDIE(dice roller), DUNGEON GRAFIX I, DUNGEON GRAFIX II, and FANTASY PEOPLE for advanced players. \$6 on your C-30, or \$8 on his. no listings.

BOWLING, No. of pins hit and score shown \$5 on your tape or \$6 on his. Add \$2 and either HANGMAN or CHECKERS will be put on other side. Listing for half price. John Collins, 713 Bradford Dr. Ft. Walton Beach, FL 32548.

Bob Weber is part of W&W Software Sales, 6594 Swartout Rd. Algonac, MI 48001 and has reorganized their software. They now have 4 cassettes with 5 programs each for \$10 each in a preprogrammed form. Send for complete list/descriptions.

PROGRAMMABLE KEYBOARD? The latest delay hinges on a Texas Instruments petition now before the FCC. They have asked for a change in the procedures by which computer/parts are approved for use. Apparently RCA tried this two years ago but never pushed very hard. The current procedure is for a computer manufacturer to have his whole system approved as meeting RFI, TVI, etc., limits, while the proposal is to have only the RF modulator inspected/approved. This has thrown the manufacturers into a tizzy as they feel that if the rules or standards are changed in the middle of the stream they'll have design and manufacturing costs to contend with. Bally, who haven't gone into production yet, feels they have to wait and see which way the FCC goes to avoid the 50-50 chance of being in error when the decision is made. For a company that thrives on gambling, these are unacceptable odds. Here is an excerpt from Consumer Electronics monthly 5/79 p. 31:

Doubt about future standards for personal computers will delay product introductions as well as halt production of units introduced, but not yet in production. Bally, for one, will hold off manufacturing its upgraded game/computer introduced at Winter CES until a decision is reached.

"We have two units to go with," says national sales manager, Jack Nieman, "the one we showed at CES for around \$650 and a lower-priced unit for around \$350. But it could cost us millions of dollars if we make a decision on which unit to go with before the FCC makes a decision." □

My discussions with Bally indicate that they expect a corporate discussion/decision around January, with delivery months later than that. Needless to say, the whole situation is disappointing and frustrating. Fortunately, we have some people working on ways to 'make our own', and we can see a little glimmer of light, as reported earlier this issue. Subscribers who are working on a keyboard, memory addition, or any other "geodie" are urged to write me so that I can build up a team to get the needed hardware built and available for us all.

-54-

## ARCADIAN

Robert Fabris, stapler  
3626 Morrie Dr.  
San José, CA 95127



94550HUS0635L  
R. HAUSER  
635 LOS ALAMOS AVENUE  
LIVERMORE, CA

94550

FIRST CLASS